REMARKS

Upon entry of this paper, pending in this application are claims 16-46, of which claims 16, 33, 45 and 46 are independent.

Claim Rejections - 35 U.S.C. §102

Claims 16 and 45 are rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 5,481,741 ("McKaskle"). Applicant respectfully traverses this rejection for the following reasons.

Claim 16 recites a method of *mapping graphical block diagram block parameters* in a graphical block diagram modeling environment. In the method of the claimed invention, a user-defined block parameter is received and *processed* to *optimally* produce a run-time block parameter. Claim 45 is a medium claim that parallels claim 16.

Applicant submits that McKaskle fails to disclose each and every element of the claimed invention. Applicant submits that McKaskle does not disclose receiving and processing the user-defined *block parameter* to optimally produce a run-time *block parameter*, as recited in claim 1. The Examiner cites column 5, lines 47-51, column 6, lines 24-27 and column 11, lines 57-62 to support the statement that this element of the claimed invention is disclosed by McKaskle. Applicant respectfully disagrees that McKaskle so discloses. McKaskle, including the cited references, is directed to an different invention. McKaskle addresses visualization of front panel controls or indicators, by disclosing how to use attributes of those controls in a data flow diagram or programatically change attributes of indicators by the data flow diagram. See, e.g., Col 5, lines 62-67 ("more meaningful visual output to the user"; "attribute node to affect the visual output of a control provided on the front panel . . ."). [McKaskle equates parameters and attributes of controls or indicators. Id.] Claims 16 and 45 claim an invention that is directed to how parameters associated with blocks in an executable block diagram are used during exection. These claims are not directed to front panel controls or indicators.

McKaskle discloses in Fig. 3 a virtual instrument (40) that includes a front panel (42) and a block diagram (46) created using a front panel editor (36) and a block diagram editor (30), respectively. The front panel (42) includes graphical representations of input and output variables provided to the virtual instrument (40), which are referred to as *controls* and *indicators*, respectively. The block diagram (46) provides a visual representation of a procedure or method by which a specified value for an input variable displayed in the front panel (42) can produce a corresponding value for an output variable in the front panel (42). See McKaskle, column 12, lines 8-40.

Applicant submits that the attributes or parameters of the control or indicator in the front panel disclosed in McKaskle do not correspond to the *graphical block diagram block parameters* recited in the claimed invention. McKaskle discloses accessing and changing the attributes of the control or indicator in the front panel (42), not the parameters of blocks themselves in the block diagram (46). McKaskle does not disclose receiving and processing the parameters of the blocks in the block diagram (46).

Additionally, Applicant submits that McKaskle does not disclose *processing* the user-defined block parameter to *optimally* produce a run-time block parameter, as recited in the claimed invention. The claimed invention *processes* the user-defined block parameters to optimally produce a run-time block parameter.

McKaskle discloses an attribute node for containing one or more attributes of the control or indicator. In McKaskle, the attribute node allows two types of operations, reading an attribute node or writing to an attribute node. See McKaskle, Abstract. In McKaskle, the user-defined attributes of the controls or indicators are written to the attribute node and the attribute node simply contains the user-defined attributes of the controls or indicators. McKaskle discloses that the attributes contained in the attribute node can be read and used without being processed.

Furthermore, the claimed invention processes the user-defined block parameter to *optimally* produce a run-time block parameter. In the claimed invention, the optimal implementation of a run-time parameter can be achieved by mapping a user-defined block parameter to a run-time parameter. The mapping of the user-specified parameter to the run-time parameter enables a graphical block diagram tool to produce optimal implementations of the block equations for a given user supplied data set. See the pending application, page 3, lines 22-28.

In comparison, McKaskle discloses that a user can programmatically access various attributes of the control or indicator in the front panel (42). McKaskle also discloses that the user can make changes that affect the output of or appearance of the control or indicator in the front panel (42) during the execution of the block diagram (46). See McKaskle, column 5, lines 47-51. McKaskle discloses that the changes can be written to the attribute node to update the attribute of the controls or indicators in the front panel. McKaskle, however, does not disclose processing the user-defined block parameters to *optimally* produce a run-time block parameter.

In light of the foregoing reasons, Applicant submits that McKaskle fails to disclose each and every element of claims 16 and 45. Applicant therefore requests the Examiner to reconsider and withdraw the rejection of claims 16 and 45 under 35 U.S.C. §102(e), and pass the claims to allowance.

Claim Rejections - 35 U.S.C. §103

Claims 17-20, 22-26, 33-38 and 46 are rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over McKaskle in view of U.S. Patent No. 6,754,881 ("Dardinski"). Applicant respectfully traverses the rejection for the following reasons.

No Motivation to Combine McKaskle and Dardinski

Applicant submits that there is no motivation to combine the teachings of the cited prior art references. McKaskle relates to a data flow diagram and teaches attributes nodes for the controls and indicators in the front panel. Dardinski relates to a process control configuration system and teaches controlling object appearance in

the data process control configuration system. Applicant submits that McKaskle and Dardinski teach different subject matter and hence there is no motivation to combine the teachings of McKaskle and Dardinski.

To comply with the requirements of a *prima facie* case of obviousness, specific findings should be made as to the motivation that those of ordinary skill in the art would have selected the cited references for combination in the claimed manner without the knowledge of the claimed invention. *In re Kotzab*, 217 F.3d 1365, 1371, 55 USPQ2d 1313, 1317 (Fed. Cir. 2000). The Examiner, however, simply asserts in the Office Action that the combination of McKaskle and Dardinski would enhance the prior art system. Applicant submits that the motivation to combine the teachings of the cited prior art references should be based on objective evidence of record. *In re Lee*, 277 F.3d 1338, 1342-44, 61 USPQ2d 1430, 1433-34 (Fed. Cir. 2002). Applicant submits that the Examiner's assertion is subjective, but not objective. "[T]he factual question of motivation is material to patentability, and could not be resolved on subjective belief and unknown authority." *Id.*

In light of the foregoing reasons, Applicant submits that there is no motivation to combine the teachings of McKaskle and Dardinski. Applicant therefore requests the Examiner to reconsider and withdraw the rejection of claims 33-38 and 46 under 35 U.S.C. §103(a), and pass the claims to allowance.

McKaskle and Dardinski Not Teach All Limitations

Applicant submits that McKaskle and Dardinski, even if combined, fail to teach or suggest all of the limitations of the claimed invention.

A. Claims 17-20 and 22-26

Claims 17-20 and 22-26 depend upon independent claim 16 and add limitations to claim 16. Applicant submits that McKaskle and Dardinski do not teach **processing the user-defined block parameter to optimally produce a run-time block parameter**, as recited in claim 16.

Dardinski is cited by the Examiner to provide teachings for the limitations added in dependent claims. Dardinski teaches appearance objects (or other data and/or programming constructs) defining the appearance of configurable system components in graphical editors or other views in which the components may be depicted. Dardinski also teaches the configurable objects can be used to define blocks, loops and other components of a process control system. Dardinski, however, does not teach processing the user-defined block parameter to optimally produce a run-time block parameter, as recited in claim 16.

Additionally, Applicant submits that McKaskle and Dardinski do not teach inversely mapping the block-run-time parameters to the user-defined block parameter to optimize block implementation, as recited in claim 17. The Examiner cites Dardinski at column 17, lines 42-52 and Fig. 13, as teaching this limitation. Applicant respectfully disagrees. Dardinski teaches the relationship between a Typed Object and an instance of an Object Type, and the relationship between the instance of the Object Type and a Parameterized Object which defines the Typed Object. That is, Dardinski teaches that a Typed Object references an Object Type, which references a Parameterized Object. Dardinski provides an example that if a user drags a symbolic representation of a definition (a block) and drops it into a view, this relationship provides quick access to the Parameterized Object which actually defines the block.

In contrast, the claimed invention inversely maps the block-run-time parameters to the user-defined block parameter to optimize block implementation. In an illustrative embodiment described at page 10, line 26 through page 11, line 10 of the pending application, a parameter processor (94) makes a call to the blocks to set up run-time parameters. In response, the blocks define run-time parameters and map the run-time parameters to the user-defined parameters. Dardinski does not teach the mapping of the run-time parameters to the user-defined parameters. Rather, Dardinski just teaches that a symbolic representation of a definition (a block) references the Parameterized Object which actually defines the block. .

Additionally, Applicant submits that McKaskle and Dardinski do not teach mapping by discarding at least a portion of the plurality of user-defined block parameters to reduce memory requirements, as recited in claim 22. The Examiner cites Dardinski at column 82, line 23 as teaching this limitation. Applicant respectfully disagrees. Dardinski teaches general graphical control algorithm diagram editor functions, including graphical functions and database functions. Dardinski also teaches that the database functions can add and delete blocks to sheet: Dardinski, however, does not teach discarding at least a portion of the plurality of user-defined block parameters to reduce memory requirements. Dardinski only teaches deleting blocks.

Additionally, Applicant submits that McKaskle and Dardinski do not teach pooling together like non-interfaced run-time block parameters to reduce repetition of the non-interfaced run-time block parameters, as recited in claim 23. The Examiner cites Dardinski at column 11, lines 1-5 as teaching this limitation. Applicant respectfully disagrees. Dardinski teaches that users can organize parameters into groups, that each group contains logically-related parameters, and that the groups can be pre-defined and/or defined by the user. In contrast, the claimed invention programmatically pools together like non-interfaced run-time block parameters to reduce repetition of the non-interfaced run-time block parameters. Dardinski does not teach pooling together like non-interfaced run-time block parameters.

In light of the foregoing reasons, Applicant submits that McKaskle and Dardinski fail to teach all of the limitations of claim 16. Claims 17-20 and 22-26, which depend upon claim 16, are not rendered obvious over the cited prior art references because these dependent claims incorporate the patentable features of independent claim 16 that are not obvious over the combination of the prior art references. Applicant therefore requests the Examiner to reconsider and withdraw the rejection of claims 17-20 and 22-26 under 35 U.S.C. §103(a), and pass the claims to allowance.

B. Claims 33-38 and 46

Claim 33 recites a method of mapping graphical block diagram block parameters in a graphical block diagram modeling environment. In the method of the claimed invention, user-defined block parameters is received and processed to optimally produce run-time block parameters. Like non-interfaced run-time block parameters are pooled together to create a run-time parameter expression for use during modeling. Claim 46 is a medium claim that parallels claim 33.

Applicant submits that cited prior art references fail to teach or suggest all of the limitations of the claimed invention. Applicant submits that McKaskle and Dardinski do not teach processing user-defined block parameters to optimally produce run-time block parameters, as recited in claim 33. McKaskle teaches that a user can programmatically access various parameters of *the control or indicator* in the front panel. Dardinski teaches defining parameters of the appearance objects by the users. McKaskle and Dardinski, however, do not teach processing the user-defined block parameters to optimally produce run-time block parameters.

Additionally, Applicant submits that McKaskle and Dardinski do not teach pooling together like non-interfaced run-time block parameters to create a run-time parameter expression for use during modeling, as recited in claim 33. Dardinski is cited by the Examiner to provide teachings for this limitation.

Dardinski teaches that parameters are organized into groups and each group contains logically-related parameters. Dardinski also teaches that the groups can be pre-defined and/or defined by the user. Dardinski, however, does not teach pooling together like non-interfaced run-time block parameters to create a run-time parameter expression for use during modeling, as recited in claim 33.

In light of the foregoing reasons, Applicant submits that McKaskle and Dardinski fail to teach all of the limitations of claims 33 and 46. Claims 34-38, which depend upon claim 33, are not rendered obvious over the cited prior art references. Applicant therefore requests the Examiner to reconsider and withdraw the rejection of claims 33-38 and 46 under 35 U.S.C. §103(a), and pass the claims to allowance.

Claim Rejections - 35 U.S.C. §103

Claims 21, 27-29, 30-32 and 39-44 are rejected under 35 U.S.C. §103(a) as being unpatentable over McKaskle in view of Dardinski and further in view of U.S. Patent No. 5,966,532 ("McDonald"). Applicant respectfully traverses the rejection for the following reasons.

Claims 21, 27-29 and 30-32 depend upon claim 16 and claims 39-44 depend upon claim 33.

Applicant submits that cited prior art references fail to teach or suggest all of the limitations of the claimed invention. Applicant submits that McKaskle, Dardinski and McDonald do not teach processing the user-defined block parameter (parameters in claim 33) to optimally produce a run-time block parameter (parameters in claim 33), as recited in claim 16.

McDonald is cited by the Examiner to provide teachings for the limitations added in dependent claims. McDonald relates to generating graphical code in a graphical programming system, more specifically a data flow diagram is generated from a wizard associated with a separate front panel. McDonald teaches the graphical programming system includes a plurality of front panel objects or controls which represent the user interface. McDonald also teaches that the user can select parameter values in the wizard to configure certain aspects of the graphical code being created. Col. 11, lines 42-55. McDonald further teaches that the graphical code generation wizard selects a graphical code template in response to the control and configures the graphical code template with the parameter values. McDonald, however, does not teach processing the user-defined block parameter to optimally produce a run-time block parameter, as recited in claim 16.

Additionally, Applicant submits that McKaskle, Dardinski and McDonald do not teach pooling together like non-interfaced run-time block parameters to create a run-time parameter expression for use during modeling, as recited in claim 33. Dardinski is cited by the Examiner to provide teachings for this limitation.

Dardinski teaches that parameters are organized into groups and each group contains logically-related parameters. Dardinski also teaches that the groups can be pre-defined and/or defined by the user. Dardinski, however, does not teach programmatically pooling together like non-interfaced run-time block parameters to create a run-time parameter expression for use during modeling, as recited in claim 33.

In light of the foregoing reasons, Applicant submits that McKaskle, Dardinski and McDonald fail to teach all of the limitations of claims 16 and 33. Claims 21, 27-29, 30-32 and 39-44, which depend upon one of claims 16 and 33, are not rendered obvious over the cited prior art references. Applicant therefore requests the Examiner reconsider and withdraw the rejection of claims 21, 27-29, 30-32 and 39-44 under 35 U.S.C. §103(a), and pass the claims to allowance.

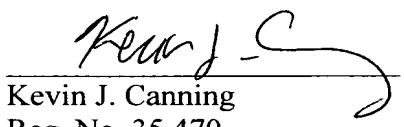
Conclusion

In view of the foregoing, it is respectfully submitted that this application is now in condition for allowance. Should there be any outstanding issues of patentability following the entry of this response, a telephone interview is respectfully requested to resolve such issues.

Please charge any shortage or credit any overpayment of fees to our Deposit Account No. 12-0080. Any fee due is authorized to be charged to the aforementioned Deposit Account.

Respectfully submitted,

LAHIVE & COCKFIELD, LLP



Kevin J. Canning
Reg. No. 35,470
Attorney for Applicant
28 State Street
Boston, MA 02109-1784
Tel: (617) 227-7400
Fax: (617) 742-4214

Date: May 17, 2005